UNIVERSITÉ DE MONTRÉAL

CENTRE DE RECHERCHE SUR LES TRANSPORTS

MÉMOIRE D'ÉTUDIANT

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC, H3C 3J7

PROBLÈME D'HORAIRE D'ÉQUIPAGE D'UNE COMPAGNIE D'AVIATION

par

Anca Paskievici

Ce rapport de recherche a été publié grâce à une subvention du Fonds F.C.A.C. pour l'aide et le soutien à la recherche.

Université de Montréal

Centre de recherche sur les transports - Publication #374
Département d'informatique et de recherche opérationnelle
Avril 1984

dépôt légal - 3e trimestre 1984

TABLE DES MATIÈRES

SOMMAIRE iv

CHAPITRE 1 - INTRODUCTION 1

1.1 DÉFINITION DU PROBLÈME 2

1.1.1 Définitions 2

1.1.2 Contraintes 4

1.1.3 Coûts 5

1.2 MÉTHODES DE RÉSOLUTION 6

CHAPITRE 2 - ALGORITME D'INSERTION PARALLÈLE 13

2.1 PROBLÈMES DE "ARC ROUTING" AVEC CAPACITÉ 13

2.2 CONSTRUCTION DU RÉSEAU 14

2.3 ALGORITHME D'INSERTION PARALLÈLE 20

2.3.1 Algorithme adapté 20

2.3.2 Domaine aérien vs domaine scolaire 25

2.4 EXPÉRIMENTATION 27

CHAPITE 3 - ALGORITHME PROPOSÉ 40

3.1 PROCÉDURE 40

3.1.1 Phase de construction de segments 41

3.1.2 Phase de formation de rotations 43

3.1.3 Phase d'amélioration 50
3.2 EXPÉRIMENTATION 54
3.2.1 Premier ensemble de données 54
3.2.2 Deuxième ensemble de données 56

CHAPITRE 4 - CONCLUSION 64

RÉFÉRENCES 66

REMERCIEMENTS 69

SOMMAIRE

Dans ce rapport, nous nous intéressons à l'une des parties du problème d'horaire d'équipage d'une compagnie d'aviation, soit à la génération d'un ensemble de rotations.

Notre objectif est de mettre au point une méthode simple qui génère un bon ensemble de rotations. Cette solution pourra par la suite, être améliorée par des packages basés sur la méthode de Rubin(utilisée actuellement dans des compagnies) ou servir d'initialisation à des méthodes basées sur la programmation mathématique, comme par exemple celle de Crainic.

Suite au succès remporté par Bodin lors de l'utilisation d'algorithmes de type "routing" pour résoudre le problème, nous étudions l'application d'un algorithme de même type, soit un algorithme d'insertion parallèle utilisé avec succès dans le contexte du transport scolaire. L'étude de cette méthode nous amène à suggérer un nouvel algorithme heuristique pour la résolution du problème de la génération de rotations.

CHAPITRE 1

INTRODUCTION

Le problème d'horaire d'équipage d'une compagnie d'aviation est d'assigner aux équipages des suites de vol ou "rotations" à effectuer. Cette assignation doit se faire de manière à réduire les coûts, tout en respectant les divers réglements.

Le problème se décompose en deux phases principales :

- -la génération de rotations.
- -la création de blocs.

La première phase du problème consiste à créer un ensemble de rotations contenant tous les vols inscrits au plan de vol des avions de la compagnie. Par la suite, la deuxième phase du problème regroupe les rotations en blocs constituant l'horaire mensuel d'un équipage.

La première phase est une étape très importante du problème parce qu'elle peut permettre de réaliser des économies appréciables. Cette phase est également difficile à effectuer étant donné les nombreuses contraintes fixées sur les rotations. Nous allons donc nous intéresser à cette phase, soit au problème de la génération d'un ensemble de rotations.

1.1 DÉFINITION DU PROBLÈME

Afin de définir le problème, nous présentons les définitions des diverses notions utilisées dans le domaine aérien, ainsi que les principales contraintes et la structure des coûts associées au problème.

1.1.1 Définitions

. *Équipage* : le pilote et le(s) co-pilote(s) nécessaires pour un vol.

. *Vol* : un vol d'une ville à une autre, à un jour et une heure donnés.

. *Vol siamois* : un vol sur lequel plus d'un équipage est alloué (*).

. *Base d'opération* : ville de résidence des équipages.

. *Nuit* : période de repos qu'un équipage doit prendre en un endroit différent de sa base d'opération.

(*) Le(s) équipage(s) supplémentaire(s) prennent le vol comme passagers afin de continuer leur service plus loin. Cette situation est acceptée par certaines compagnies lorsqu'elle est économique.

. *Temps de vol* : période de temps entre le départ et l'arrivée d'un vol.

. *Période de service* : période de temps durant laquelle un équipage effectue une suite de vols sans période de repos.

. *Temps de service* : période de temps entre le début de la période de préparation du premier vol et la fin de la période de compte-rendu du dernier vol d'une période de service.

. *Rotation* : voyage composé d'une suite de vols, entrepris par un équipage, qui commence et se termine à la base d'opération de cet équipage, tout en respectant les règles de sécurité, la convention collective et la politique de la compagnie.

. *Temps total de voyage* : période de temps entre le début de la période de préparation de la première période de service et la fin de la période de compte-rendu de la dernière période de service d'une rotation.

. *Bloc* : suite de rotations constituant l'horaire du mois d'un équipage.

1.1.2 Contraintes

Le processus de génération d'un ensemble de rotations est sujet à de nombreuses contraintes découlant des règlements de sécurité, des politiques de la compagnie, et de la convention collective. Les contraintes les plus restrictives, provenant du contrat collectif des pilotes, se situent au niveau des temps de vol, de service, d'attente, et de repos :

- durée maximale du temps de service.
- durée maximale du temps de vol pour une période de service.
- durée minimale du temps de correspondance entre vols de même avion (TCA).
- durée minimale du temps de correspondance entre vols d'avion différent (TC).
- durée minimale d'une nuit.

Mentionnons également le problème des vols à exceptions de fréquence. Dans leur horaire, la plupart des vols apparaissent chaque jour de la semaine. Or, il arrive que pour certains jours des vols ne s'effectuent pas ou comportent plus ou moins d'escales. Ces vols ont alors des exceptions de fréquence.

1.1.3 Coûts

La plupart des compagnies d'aviation nord-américaines ont un système complexe d'attribution des coûts aux rotations. La compagnie doit créditer du temps de vol supplémentaire(en plus du temps de vol effectué) lorqu'une rotation présente certaines situations désagréables(comme par exemple un temps d'attente trop long). De plus, afin de simplifier les calculs, le coût d'une nuit sera estimé en temps de vol.

Le coût d'une rotation de n périodes de service est évalué par le temps de vol crédité(HC) calculé de la manière suivante:

$$HC = MAX\{ TV/\lambda_{TV}, \sum_{i=1}^{n} \max\{ TP_i/\lambda_{TP}, TVL_i, MVL \} \} + (n-1)CR$$

où

TV = temps total de voyage.

λ_{TV} = coefficient de garanti pourTV.

TP_i = temps total de service pour la $i^{ème}$ période de service.

λ_{TP} = coefficient de garanti pour TP.

TVL_i = temps de vol pour la $i^{ème}$ période de service.

MVL = temps de vol minimum crédité par période de service.

CR = coût d'une nuit à l'hôtel estimé en temps.

Nous constatons qu'une telle attribution des coûts est non-linéaire. Les rotations ne peuvent donc être estimées qu'une fois complètes, étant donné l'impossibilité d'attribuer des coûts aux vols.

Le problème de génération d'un ensemble de rotations est de trouver, à partir de l'horaire des vols de la compagnie, un ensemble de rotations à coût minimal qui contient tous les vols, tout en respectant les contraintes.

1.2 MÉTHODES DE RÉSOLUTION

La formulation du problème la plus répandue est celle d'un problème de recouvrement d'ensemble("set covering"). Nous cherchons ainsi un ensemble de rotations légales recouvrant tous les vols avec un coût minimal. L'expression d'un tel problème est la suivante :

$$\text{Minimiser} \quad \sum_{j=1}^{n} c_j x_j$$

$$\text{sujet à} \quad \sum_{j=1}^{n} a_{ij} x_j \geq 1 \quad , \quad i=1,\ldots,m.$$

où

n = nombre de rotations légales.

m = nombre de vols.

c_j = coût de la rotation j.

et

$$x_j = \begin{cases} 1 \text{ si la rotation } j \text{ est dans la solution.} \\ 0 \text{ sinon.} \end{cases}$$

$$a_{ij} = \begin{cases} 1 \text{ si le vol } i \text{ est recouvert par la rotation } j. \\ 0 \text{ sinon.} \end{cases}$$

Notons qu'ici, les vols siamois sont permis(un vol peut être recouvert par plus d'une rotation). Dans le cas où de telles situations ne sont pas permises, il suffit de transformer dans l'équation de contrainte le "$\geq$" par un "=". Le problème de recouvrement d'ensemble devient alors un problème de partition d'ensemble("set partitioning"). Notons également que certaines contraintes supplémentaires peuvent être ajoutées au problème, comme par exemple des contraintes limitant le nombre de pilotes ou d'heures de vol/pilote disponibles à chaque base d'opération.

Cette formulation nécessite la génération de toutes les rotations réalisables(colonnes de la matrice des contraintes). Or, de par la nature combinatoire du problème, le nombre de rotations légales s'accroit très rapidement avec le nombre de vols(il peut atteindre le million pour un réseau continental d'une compagnie nord-américaine). Donc, pour pouvoir traiter de grands réseaux, des méthodes de réduction doivent être appliquées à l'ensemble des rotations réalisables afin de le réduire en un ensemble de dimension raisonnable. Il existe des méthodes de réduction de type exact qui utilisent des règles mathématiques, assurant alors l'obtention d'un sous-ensemble de rotations réalisables contenant la solution optimale. Or, il s'avère que ces méthodes ne réduisent que de peu la taille du problème. Il faut donc avoir recours aux méthodes de réduction de type heuristique qui éliminent les rotations semblant mauvaises(en tant que longueur des rotations par exemple). Ces

méthodes sont efficaces pour la réduction du problème mais risquent d'éliminer des rotations qui, malgré leur apparence, impliqueraient une solution optimale.

Étant donné un ensemble de rotations réalisables de dimension raisonnable, plusieurs techniques exactes peuvent être appliquées :

- des méthodes de coupe par hyperplans
(Gomory(13), Glover(11,12), Young(23), Balas(3), Bellmore(5), Salkin(20), Balas & Ho(4)).

- des méthodes énumératives
(Balas(2), Geoffrion(10), Pierce(17), Garfinkel & Nemhauser(9), Lemke, Salkin & Spielberg(15), Pierce & Lusky(18), Heurgon(14), Marsten(16), Etcheberry(8)).

- des méthodes se basant sur la théorie des groupes
(Shapiro(21), Thiriez(22)).

Comme généralement le nombre de rotations légales dépasse les limites des algorithmes existants, les méthodes de réduction mathématiques sont peu efficaces, et les méthodes de réduction heuristiques ne garantissent pas l'optimalité de la solution, il est extrêmement difficile d'obtenir un ensemble restreint de rotations réalisables contenant la solution optimale du problème de recouvrement associé.

Une approche totalement différente est celle d'amélioration successive présentée par Rubin(19). Plutôt que d'essayer de générer un sous-ensemble de "bonnes" rotations, cette méthode améliore pas à pas une solution initiale donnée en résolvant de petits problèmes de recouvrement. Nous présentons une brève description de la méthode :

Supposons S une solution initiale donnée.

Soient

$S = \{ P_i , i=1,\dots,r\}$ = ensemble de r rotations légales(colonnes).

$F_i = F(P_i)$ = ensemble des vols(lignes) recouverts par P_i.

$C_i = C(P_i)$ = coût de la rotation P_i, $C_i > 0$.

$B_i = B(P_i)$ = base d'opération où la rotation P_i débute ;
$1 \leq B_i \leq n$ (les bases sont indexées séquentiellement).

$H_i = H(P_i)$ = nombre d'heures de vol/pilote par mois nécessaire à la rotation P_i ; $H_i \neq 0$.

F = ensemble de vols.

n = nombre de bases d'opération.

Puisque S est une solution, nous avons :

$$\cup_i F_i = F$$
$$F_i \cap F_j = \emptyset \quad \text{si } i \neq j$$
$$H_j^L \leq \sum_{i \neq j, F_i \neq j} H_i \leq H_j^U \quad , j=1,\ldots,n.$$

Sans perte de généralité, nous supposons $C_i \geq C_j$ si $i \geq j$. L'idée de base de la méthode est de choisir des sous-ensembles $T \subset S$ se composant de une ou plusieurs colonnes et de générer toutes les rotations légales recouvrant seulement les lignes dans $F_T = \cup_{P_i \in T} F_i$. Nous avons ainsi un problème miniature de recouvrement d'ensemble impliquant seulement les lignes de F_T. Les colonnes générées seront celles de T et toutes les autres possibilités légales. Une borne supérieure du coût de la solution optimale de ce problème est $C = \sum_{P_i \in T} C_i$. Si le coût de la solution optimale T' est inférieur à C, nous pouvons remplacer T par T' dans S, obtenant ainsi une solution améliorée S' au problème initial. Un nouveau sous-ensemble T est choisi et le processus est répété.

Notons que, pour cette méthode, la qualité de la solution finale est directement proportionnelle à celle de la solution initiale, d'où l'importance du choix d'une bonne solution de départ.

Mentionnons également qu'il est possible de résoudre le problème par un algorithme de génération de colonnes qui génère les rotations si nécessaire. Cet algorithme proposé par Crai-

nic(7) génère d'abord un ensemble de rotations. Le problème de recouvrement d'ensemble est ensuite résolu linéairement et les variables duales sont utilisées pour générer un ensemble d'autres rotations de coût réduit négatif qui sont ajoutées au problème original. Ce processus est répété jusqu'à ce que l'amélioration de la fonction objective soit petite. Un algorithme heuristique est alors appliqué pour obtenir une solution entière. Notons que l'application d'un tel algorithme est relativement complexe.

Pour résoudre de très grands problèmes d'horaire d'équipage d'avion, les algorithmes heuristiques restent encore les seuls recours. Nous présentons, dans cette section, un tel algorithme développé par Baker & Bodin(1). La méthode se décompose en trois phases : (1) utilisation d'un générateur de rotations, (2) application d'un algorithme heuristique de recouvrement d'ensemble utilisé pour obtenir une solution initiale, (3) utilisation de plusieurs techniques d'amélioration qui réduisent le coût de la solution. Le générateur de rotations énumère des rotations respectant les contraintes. Dans cette énumération, des procédures de réduction de type heuristique sont incorporées. L'algorithme heuristique débute avec un ensemble de solution vide auquel il ajoute la rotation qui minimise le coût par vol non recouvert. Le processus est répété jusqu'à ce que tous les vols soient recouverts. Trois techniques d'amélioration sont ensuite appliquées. La première est une procédure "2-opt" étendue qui échange les séquences de vols de

rotations. Si k rotations voyagent à une même ville, un problème d'allocation kxk peut être résolu afin de déterminer la configuration optimale des rotations. La seconde technique d'amélioration est une procédure de "fusion". Étant donné un ensemble de solutions réalisables, les deux premières solutions sont comparées afin de savoir si une solution mixte de coût réduit existe. La partition de coût minimal résultante est alors comparée à la troisième solution en vue d'amélioration. Le processus est répété jusqu'à ce que toutes les solutions aient été utilisées. La troisième technique d'amélioration joint ensemble deux rotations de même base. Ainsi, une rotation peut traverser sa base et s'y terminer qu'à son deuxième passage. Notons que l'application de cette méthode sur les données de Federal Express Corporation génère une solution relativement bonne lors de l'existence d'une seule base, tandis que les résultats sont moins concluants lorsque plusieurs bases sont impliquées.

Nous tentons de remédier à ce problème en testant, dans le prochain chapitre, l'application d'un autre algorithme de même type. Cet algorithme développé pour le domaine du transport scolaire, génère avec succès des routes de véhicules équilibrées. Nous adaptons alors cet algorithme au domaine aérien et étudions son comportement. Cette étude nous amène à proposer, dans le chapitre succédent, un nouvel algorithme heuristique pour résoudre le problème de la génération de rotations.

CHAPITRE 2

ALGORITHME D'INSERTION PARALLÈLE

Dans ce chapitre, nous étudions la résolution du problème par un algorithme d'insertion parallèle développé pour les problèmes de "arc routing" avec capacité. À cette fin, nous présentons d'abord la formulation du problème d'horaire comme un problème de "arc routing" et nous construisons le réseau correspondant. Par la suite, nous développons une adaptation de l'algorithme à notre problème. Nous présentons alors les résultats de cet algorithme et nous introduisons séquentiellement une série de modifications en vue d'en améliorer la solution. Nous concluons finalement sur les difficultés d'application de certains principes d'insertion parallèle au problème de génération de rotations.

2.1 PROBLÈMES DE "ARC ROUTING" AVEC CAPACITÉ

Les problèmes de "arc routing" avec capacité s'énoncent généralement de la manière suivante :

"Trouver un ensemble à coût minimal de routes fermées (cycles) effectuées par un véhicule et passant par un dépôt, qui recouvre au moins une fois chaque arc du réseau en tenant compte des capacités."

En établissant les correspondances suivantes :

. route ⟼ rotation

. dépôt ⟼ base

. véhicule ⟼ équipage

. arc ⟼ vol

. capacité ⟼ temps de vol,

nous formulons notre problème d'horaire en un tel problème :

"Trouver un ensemble à coût minimal de rotations effectuées par un équipage et passant par une base, qui recouvre au moins une fois chaque vol du réseau en tenant compte des contraintes de temps de vol."

2.2 CONSTRUCTION DU RÉSEAU

Nous devons construire un réseau tenant à la fois compte de l'espace(emplacement des villes desservies par les vols) et du temps(horaire des vols).

Soient

$V = \{v_1, v_2, \ldots, v_n\}$ l'ensemble des n vols de la compagnie.

et

$X = \{x_1, x_2, \ldots, x_m\}$ l'ensemble des m villes desservies.

Dénotons par O_i l'heure de départ du vol v_i

et par D_i l'heure d'arrivée du vol v_i.

Tout vol v_i peut être représenté comme suit :

$$v_i = (x_j(O_i), x_k(D_i)) \qquad \begin{array}{l} 1 \leq i \leq n, \\ 1 \leq j \leq m, \\ 1 \leq k \leq m, k \neq j. \end{array}$$

où $x_j(O_i)$ représente l'heure de départ du vol v_i de sa ville d'origine x_j,
et $x_k(D_i)$ représente l'heure d'arrivée du vol v_i à sa ville de destination x_k.

Considérons le réseau suivant :
- Chaque $x_j(O_i)$ et $x_j(D_i)$ $(i=1,\ldots,n ; 1 \leq j \leq m)$ est représenté par un sommet.

- Chaque vol $v_i = (x_j(O_i), x_k(D_i))$ est représenté par un arc orienté du sommet $x_j(O_i)$ vers le sommet $x_k(D_i)$.

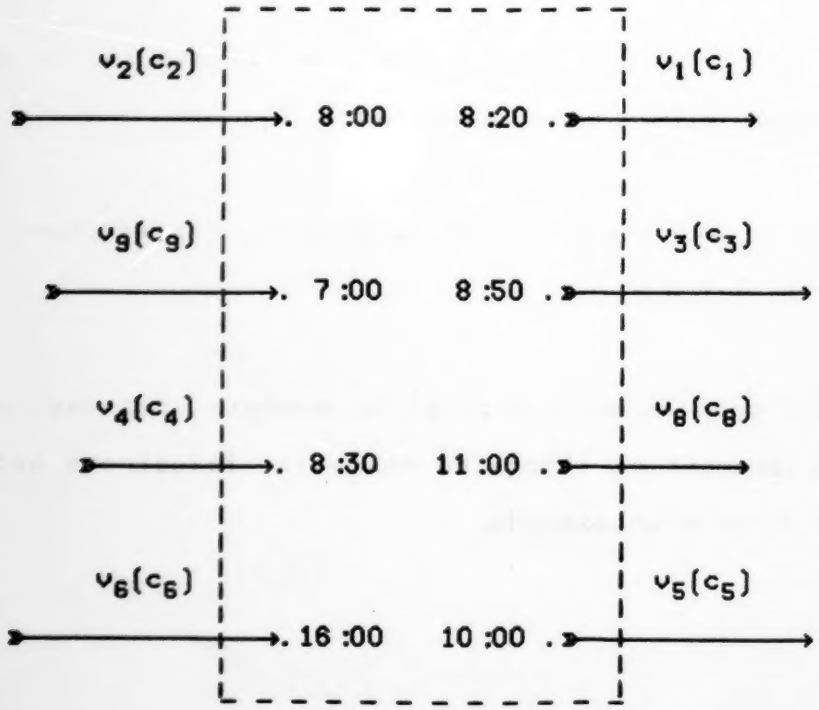
- À chaque arc v_i est associée une longueur c_i représentant le temps de vol du vol v_i.

Un tel réseau, composé de n arcs et 2n sommets, représente à la fois l'emplacement et l'horaire des vols. Illustrons cette propriété à l'aide d'un exemple.

Considérons les activités suivantes pour la ville x_5 :

vols destinations	heures d'arrivée	vols origines	heures de départ
v_2	8 :00	v_1	8 :20
v_9	7 :00	v_3	8 :50
v_4	8 :30	v_8	11 :00
v_6	16 :00	v_5	10 :00

Le réseau correspondant à x_5 est le suivant :



Nous devons maintenant compléter le réseau par des arcs inter-vols permettant aux équipages la correspondance entre vols. Nous pourrions simplement construire tous les arcs inter-vols réalisables mais, pour un réseau de grande taille, le nombre d'arcs deviendrait très élevé. Nous proposons alors de compléter le réseau de la manière suivante :

Pour chaque ville x_j composée de y_j sommets origines ($j=1,\ldots,m$) faire :

1) Trier les sommets destinations par ordre croissant d'heure d'arrivée et les sommets origines par ordre croissant d'heure de départ.

2) Ajouter :

- (y_j+1) sommets intermédiaires (les y_j premiers correspondant aux y_j sommets origines).

- un arc orienté du $i^{ème}$ sommet intermédiaire vers le $(i^{ème}+1)$ sommet intermédiaire de longueur égale à la différence d'heure entre le $(i^{ème}+1)$ et le $i^{ème}$ sommet origine, pour $i=1,\ldots,(y_j-1)$.

- un arc orienté du $y_j^{ème}$ sommet intermédiaire vers le $(y_j+1)^{ème}$ sommet intermédiaire de longueur égale à la différence d'heure entre le $y_j^{ème}$ et le premier sommet origine.

- un arc orienté du $(y_j+1)^{ème}$ sommet intermédiaire vers le premier sommet intermédiaire de longueur nulle.

- un arc orienté du $i^{ème}$ sommet intermédiaire vers le $i^{ème}$ sommet origine de longueur nulle, pour $i=1,\ldots,y_j$.

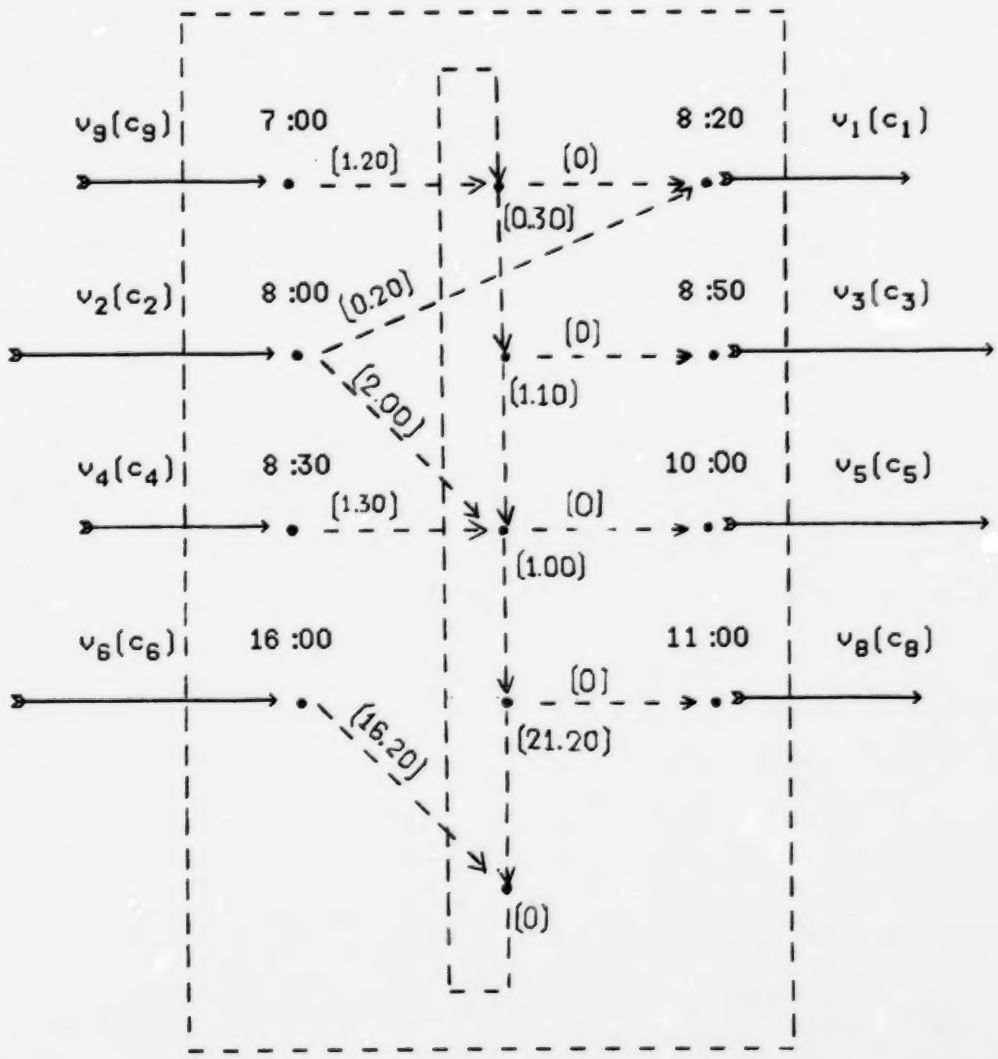
- à partir de chaque sommet destination :

. un arc orienté vers le sommet origine correspondant au vol de même avion si existant, de longueur égale au temps d'attente($\geq$ TCA).

. un arc orienté vers le sommet intermédiaire correspondant au sommet origine qui implique un temps d'attente minimal $\geq$ TC dans la même journée si possible ; sinon vers le $(y_j+1)^{ème}$ sommet intermédiaire, de longueur égale au temps d'attente.

Afin de visualiser ce réseau, reprenons l'exemple précédant correspondant aux activités de la ville x_5, en considérant que l'avion effectuant le vol v_2 effectue par la suite le vol v_1.
Supposons également que TCA = 20 minutes

et que TC = 60 minutes.



Nous avons ainsi construit un réseau espace-temps de dimension raisonnable tenant compte des contraintes de temps de correspondance, des temps de vol, et des temps d'attente.

2.3 ALGORITHME D'INSERTION PARALLÈLE

Nous présentons dans cette section une adaptation au domaine aérien de la méthode d'insertion parallèle développée pour le domaine scolaire. Nous mentionnons également quelques différences d'application de l'algorithme entre les deux domaines.

2.3.1 Algorithme adapté

Nous pouvons trouver une description de l'algorithme original dans Chapleau et al. (6). Nous adaptons l'algorithme à notre problème en modifiant quelques termes et critères employés.

L'idée de base de la méthode d'insertion parallèle est de générer plusieurs rotations en parallèle en se basant sur les deux stratégies suivantes :

1) Étant donné un vol, déterminer la rotation dans laquelle le vol devrait être inséré.

2) Étant donné une rotation, déterminer le vol libre à y être inséré.

Il est à noter que le mot "rotation" représente une rotation en construction.

L'algorithme est composé de trois phases, soient :

- une phase d'initialisation des rotations utilisant la stratégie 1) : Phase0.

- une phase de construction de rotations utilisant la stratégie 1) : Phase1.

- une phase de construction de rotations utilisant la stratégie 2) : Phase2.

PHASE0

Tantque le nombre de rotations < borne faire

début

trouver le vol libre v_l le plus éloigné ;

trouver parmi les rotations auxquelles on peut insérer v_l ,

celle qui donne un temps d'attente minimal < cte0 ;

si on ne peut pas insérer v_l

alors on crée une nouvelle rotation avec v_l

sinon on insère v_l dans la rotation

fin

PHASE1

Tantque il reste des vols non-examinés faire

début

trouver le vol libre v_l le plus éloigné non-examiné ;

trouver parmi les rotations auxquelles on peut insérer v_l ,

celle qui donne un temps d'attente minimal < cte1 ;

si on ne peut pas insérer v_l

alors v_l est examiné

sinon on insère v_l dans la rotation et tous

les vols libres deviennent non-examinés

fin ;

si aucun vol n'a été inséré

alors créer une nouvelle rotation avec le vol libre

le plus éloigné ne touchant pas à une base

PHASE2

Calculer les "ratio" des rotations ;

les rotations de ratio < moyenne des ratio sont "petites" ;

tantque il existe une rotation "petite" faire

début

trouver la rotation "petite" R_i de plus petit ratio ;

insérer dans R_i le vol donnant un temps d'attente minimal ;

R_i est "grande"

fin

où ratio = temps de vol/(temps de vol + temps d'attente).

Les principales notions utilisées dans les phases de l'algorithme sont celles de degré d'éloignement de vol, de ratio de rotation, et de temps d'attente minimal.

1) Degré d'éloignement de vol

Les Phase0 et Phase1 traitent les vols par ordre décroissant de leur degré d'éloignement que nous définissons comme suit :

.degré d'éloignement de vol = durée(en temps de vol et temps d'attente) de la plus courte rotation contenant le vol.

Ainsi, nous traitons d'abord les vols qui engendrent des rotations de longue durée. Dans le cas inverse, il résulterait sûrement de l'insertion de ces vols, une rotation encore plus longue, étant donné que la probabilité qu'aucun vol constituant leur rotation minimale ne soit pas déjà inséré ail-

leurs est très faible. Ainsi donc, nous évitons de prolonger des rotations déjà longues.

Notons que les degrés d'éloignement sont déterminés préalablement en calculant, à l'aide du réseau, la longueur des plus courts chemins de chaque vol vers chaque base et de chaque base vers chaque vol.

2) Ratio de rotation

La Phase2 tente d'améliorer les rotations de ratio inférieur ou égal à la moyenne afin que le temps d'attente ne soit pas trop élevé par rapport au temps de vol. La Phase2 peut être considérée comme une phase d'équilibre des rotations.

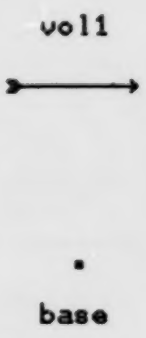
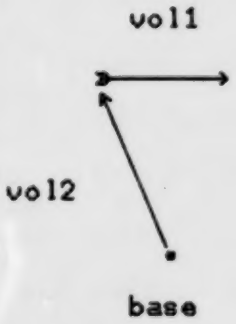
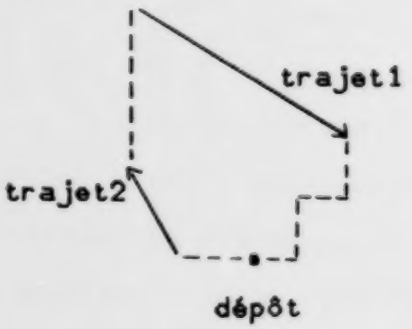
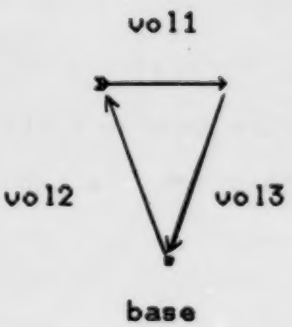
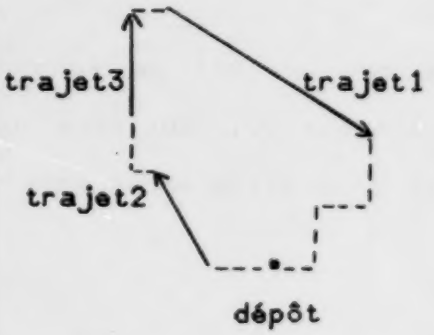
3) Temps d'attente minimal

Nous choisissons d'initialiser et de construire les rotations en minimisant le temps de corespondance entre deux vols afin d'éviter les situations désagréables où les équipages doivent attendre longtemps avant d'effectuer leur prochain vol. Les phases de l'algorithme complètent alors les rotations par le vol qui implique un temps d'attente minimal inférieur ou égal à une constante. La constante, pouvant différer d'une phase à l'autre, limite la durée du temps d'attente.

Notons que la définition du degré d'éloignement et du ratio ainsi que le critère de temps d'attente minimal ont été déterminés arbitrairement et pourront être modifiés sans pour autant changer la structure de l'algorithme.

2.3.2 Domaine aérien vs domaine scolaire

Les rotations construites par l'algorithme modifié pour le domaine aérien sont initialement composées d'un vol, ce qui implique qu'elles ne deviennent réalisables que lorsqu'elles se complètent en un cycle. Par contre, dans le domaine scolaire, l'algorithme initialise les routes non seulement par un trajet à effectuer mais également par le chemin aller-retour correspondant, entrainant ainsi la réalisabilité des routes à chaque étape. Cette différence est due au fait qu'un autobus peut revenir à son dépôt par un chemin quelconque tandis qu'un équipage d'avion ne peut retourner à sa base que par des vols. De plus, un vol n'est pas réellement "inséré" dans une rotation, mais plutôt "ajouté" avant le premier vol ou après le dernier vol de la rotation, contrairement à un trajet d'autobus qui peut être inséré entre deux trajets d'une route. Ces deux différences sont illustrées dans le tableau suivant.

	DOMAINE AÉRIEN rotation	DOMAINE SCOLAIRE route
ÉTAPE 1		
ÉTAPE 2		
ÉTAPE 3		

2.4 EXPÉRIMENTATION

Dans cette section, nous testons l'algorithme sur les données d'Air Canada. Nous présentons séquentiellement les diverses difficultés rencontrées, ainsi que les changements apportés à l'algorithme en vue de l'obtention de meilleurs résultats.

Afin d'évaluer rapidement l'algorithme, nous le testons sur un réseau restreint de DC9 d'Air Canada. Nous choisissons un ensemble de vols sans exception de fréquence, tel que le nombre de vols arrivant à une ville soit égal au nombre de vols la quittant(l'existence d'une solution réalisable est alors assurée). Nous obtenons ainsi un ensemble de 165 vols.

Les containtes utilisées pour le réseau d'Air Canada sont les suivantes :

- bases : Montréal (YUL), Toronto (YYZ), Winnipeg (YWG).
- temps de préparation pour une période de service : 60 minutes.
- temps de compte-rendu pour une période de service : 30 minutes.
- temps minimum de correspondance entre deux vols de même avion(TCA) : 20 minutes.
- temps minimum de correspondance entre deux vols d'avion différent(TC) : 60 minutes.
- temps minimum pour une période de repos : 555 minutes.
- temps maximum de service : 840 minutes.

Le coût d'une rotation de n périodes de service est déterminé par son nombre d'heures de vol créditées qui est évalué selon la formule décrite à la section 1.1.3 :

$$HC = MAX\{ TV/\lambda_{TV}, \sum_{i=1}^{n} \max\{ TP_i/\lambda_{TP}, TVL_i, MVL \}\} + (n-1)CR$$

où

$\lambda_{TV} = 3.5$

$\lambda_{TP} = 2$

MVL = 240 minutes

CR = 120 minutes

Nous évaluons les coûts des rotations en comparant le nombre d'heures de vol créditées au nombre d'heures de vol réellement effectuées.

Dans un premier temps, nous testons l'algorithme tel que décrit à la section 2.3.1 afin d'évaluer la structure des chaînes de vols obtenues. Nous ajouterons par la suite une procédure qui ferme ces chaînes en cycles afin d'obtenir des rotations réalisables.

TEST1 : Algorithme tel quel avec borne = 12, cte0 = cte1 =120 minutes.

Nous obtenons des rotations très déséquilibrées par rapport au nombre de vols les constituant. En effet, certaines rotations se composent d'une vingtaine de vols tandis que

d'autres n'en comportent qu'un seul. Nous observons que les rotations formées d'un vol qui ne se complètent pas dans la Phase1, restent également composées d'un seul vol après la Phase2. Ces rotations ne sont effectivement pas examinées dans la Phase2 puisque leur ratio est de 1(car le temps d'attente est nul) et donc supérieur ou égal à la moyenne des ratios. Afin d'éviter ce problème, nous proposons de changer la définition du ratio = temps de vol/(temps de vol + temps d'attente) par (temps de vol + temps d'attente).

TEST2 : Idem avec ratio = (temps de vol + temps d'attente).

Nous obtenons ainsi des rotations plus équilibrées. Nous choisissons alors de garder ce changement apporté à la définition du ratio.

De plus, nous remarquons que certaines rotations s'éloignent progressivement de leur base. Lorsque nous ajoutons un vol à une rotation, nous ne tenons pas compte de la direction que prend alors la rotation. Afin de diriger les rotations vers une base, nous proposons de changer le critère de "temps d'attente minimal" dans la Phase2 par celui du "chemin le plus court vers la base la plus proche". Ainsi, dans la Phase2, nous complétons une rotation par le vol libre qui se trouve sur le chemin le plus court de la rotation vers la base la plus proche.

TEST3 : Idem avec le critère du "plus court chemin vers la base la plus proche" dans la Phase2.

Nous obtenons ainsi des rotations se dirigeant vers une base, mais nous constatons que certaines rotations qui devraient être traitées en Phase2 ne le sont pas. Ceci signifie qu'il n'existe plus de chemin par lequel ces rotations peuvent revenir à une base. Ainsi, certains vols déjà utilisés par des rotations sont nécessaires à d'autres pour le retour à une base. Afin d'évaluer la fréquence de cette situation, nous permettrons les vols siamois("deadheading").

De plus, nous ajouterons à l'algorithme une phase supplémentaire(Phase3) qui complète entièrement les rotations. Cette phase se décrit de la manière suivante :

PHASE3

Les rotations ayant un temps de vol $\geq$ cte3

ou un temps total(vol+attente) $\geq$ cte3' sont "grandes" ;

Tantque il existe des rotations "grandes" R_1 faire

début

compléter R_1 par tous les vols se trouvant sur

le cycle minimal incluant R_1 et passant par une base ;

R_1 est "petite"

fin

L'algorithme devient alors le suivant :

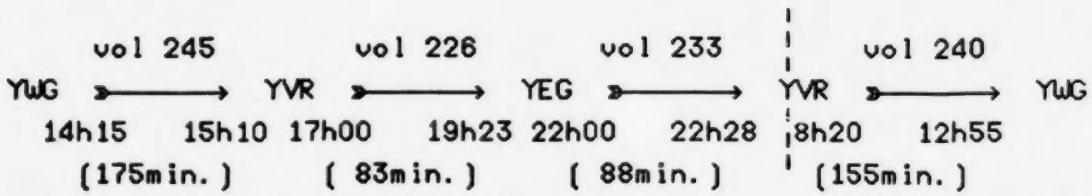
```
ALGORITHME : Début
               Phase0 ;
               Phase2 ;
               Tantque il reste des vols libres faire
                début
                 Phase1 ;
                 Phase2 ;
                 Phase3
                fin
              Fin
```

TEST4 : Idem avec le "deadheading" permis et l'ajoût de la Phase3 avec cte3 = 400 minutes, cte3' = 600 minutes.

Nous obtenons ainsi des rotations complètes mais nécessitant de nombreux "deadheading"(une dizaine de segments siamois). Cette situation se reproduit également avec des choix différents pour les valeurs des constantes. En observant les résultats, nous constatons que si certains vols initialisant des rotations avaient été mis ensemble afin de ne former qu'une rotation, le "deadheading" aurait été évité. Ainsi, en Phase0, plutôt que d'initialiser une rotation par le vol "le plus éloigné", nous devrions l'initialiser par le vol "le plus éloigné ne pouvant être utilisé au cours de la construction d'une autre rotation". Étant donné la complexité et la difficulté d'application de ce test, nous essaierons d'initialiser manuellement les rotations par des vols que nous considérons indépendants.

TEST5 : Idem avec une initialisation manuelle et sans la Phase0.

Nous obtenons des rotations sans aucun "deadheading" mais dont la plupart sont asymétriques par rapport à leur ville de repos, comme par exemple la rotation suivante :



période de service	temps de vol (en minutes)	temps d'attente (en minutes)	temps de service (en minutes)
1	346	267	703
2	155	0	245

La première période de service de cette rotation contient trois vols tandis que la seconde n'en contient qu'un seul. Cette dernière est coûteuse parce qu'elle n'a que 155 minutes de vol effectuées pour un minimum de 240 minutes de vol créditées (MVL).

Afin d'obtenir des rotations plus symétriques et moins coûteuses, nous ajoutons à l'algorithme une procédure d'amélioration(AMÉLIO) permettant aux rotations de passer plusieurs fois par la même base. Cette procédure se décrit de la manière suivante :

AMÉLIO
Tantque il reste des rotations non-examinées faire
début
déterminer la rotation R_1 non-éxaminée de temps de vol minimal lors de sa première période de service ;
trouver parmi les rotations de même base dont leur dernière période de service peut former avec la première de R_1 qu'une seule période, celle qui donne un temps d'attente minimal ;
si une telle rotation existe
alors insérer cette rotation avant R_1 ;
R_1 est examinée
fin

En regroupant ainsi deux rotations de même base en une seule rotation, nous transformons également deux périodes de service en une, sans nuit supplémentaire. Etant donné que le temps de vol maximal pour une période de service est de 420 minutes, au moins une des deux périodes de service regroupées a un temps de vol inférieur ou égal à 240 minutes. Le critère de temps de vol minimal crédité par période de service (MVL = 240 minutes) est alors effectif pour cette période. Or, en complétant la

période de service avec une autre , le temps de vol crédité en supplément ne peut que diminuer. Ainsi, nous réduisons le coût des rotations. De plus, en traitant d'abord les périodes de service de temps de vol court, nous équilibrons la durée des périodes, rendant ainsi les rotations correspondantes plus symétriques.

TEST6 : Idem avec la procédure AMÉLIO

Nous obtenons une diminution d'environ 13% du temps de vol total crédité. De plus, les rotations résultantes sont beaucoup plus symétriques. Ainsi, dans l'exemple suivant, la rotation(B) regroupée avec la rotation(A) se transforme en la rotation(C) qui, elle, est symétrique. Le temps total de vol crédité pour (A) et (B) diminue alors de 846 à 712.5 minutes :

(A)

vol 662 vol 669

YYZ ⟶ YQM ⟶ YYZ

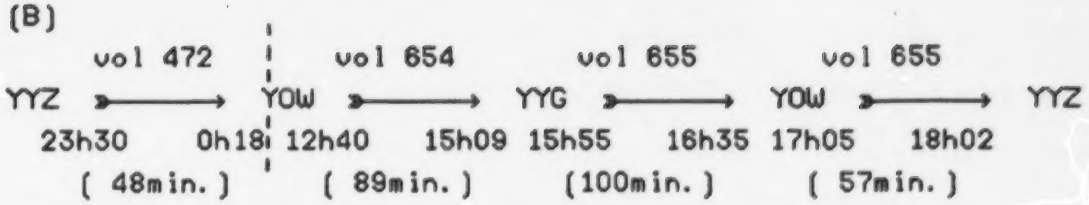
14h15 17h05 17h45 18h53

(110min.) (128min.)

période de service	temps de vol	temps d'attente	temps de service	HC_1(*)
1	238 min.	40 min.	368 min.	240 min.

avec TV = 368 minutes et HC = 240 minutes

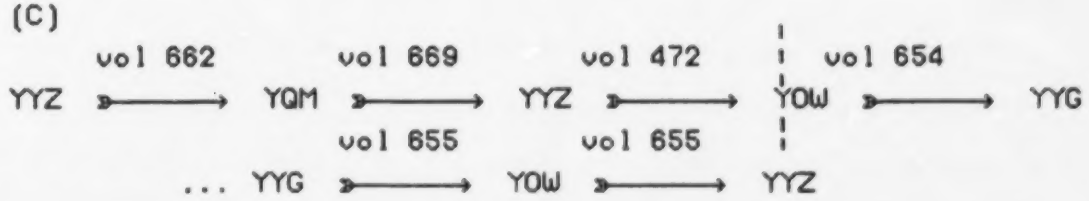
(B)



période de service	temps de vol	temps d'attente	temps de service	HC_i (*)
1	48 min.	0 min.	138 min.	240 min.
2	246 min.	76 min.	412 min.	246 min.

avec TV = 1202 minutes et HC = 606 minutes

(C)



période de service	temps de vol	temps d'attente	temps de service	HC_i (*)
1	286 min.	317 min.	693 min.	346.5 min.
2	246 min.	76 min.	412 min.	246 min.

avec TV = 1757 minutes et HC = 712.5 minutes

(*) Temps de vol crédité pour une période de service i

= $HC_i = Max\{TP_i/2, TVL_i, 240\}$ et

Temps de vol crédité pour une rotation de n périodes

= $HC = Max\{TV/3.5, \sum_{i=1}^{n} HC_i\} + 120(n-1)$

Afin d'améliorer davantage la solution, nous proposons une procédure d'échange(ECHANGE) qui échange les vols d'une période de service avec ceux d'une seconde période de service lorsqu'il en résulte une amélioration des coûts. Cette procédure se décrit comme suit :

ÉCHANGE
Déterminer toutes les suites de vol(s) contenues dans une période de service en parcourant les rotations dans l'ordre croissant de leur temps d'attente total ;
Pour chacune de ces suites faire
début
chercher une seconde suite de vol(s) de même origine et de même destination, contenue dans une période de service, en parcourant les rotations dans l'ordre décroissant de leur temps d'attente total ;
si l'échange de ces suites de vol est faisable et implique une diminition des coûts des rotations correspondantes
alors faire l'échange
fin

Ainsi, pour chaque suite de vol(s) contenue dans une période de service, nous effectuons le premier échange rencontré qui améliore les coûts. Nous considérons qu'il n'est pas nécessaire de trouver l'échange qui implique la diminution maximale des coûts étant donné que l'obtention d'un meilleur ensemble final de rotations n'est pas garanti(l'échange optimal à une

certaine étape peut empêcher de futurs échanges globalement meilleurs)

De plus, en effectuant d'abord des échanges entre une première rotation de petit temps d'attente et une seconde de grand temps d'attente, nous visons à empirer d'une certaine valeur le coût de la première en augmentant son temps d'attente et à améliorer d'une valeur supérieure le coût de la seconde en diminuant son temps d'attente. Ainsi, les échanges entre vols réduisent le coûts des rotations tout en équilibrant leur temps d'attente.

TEST 7 : Idem avec la procédure ÉCHANGE.

Notons que plusieurs emplacements de la procédure ÉCHANGE sont possibles :

- avant la procédure AMÉLIO
- après la procédure AMÉLIO
- après chaque amélioration effectuée dans la procédure AMÉLIO

Il s'avère que ce dernier emplacement fournit les meilleurs résultats. Nous obtenons alors une diminution d'environ 6% du temps de vol total crédité, à la suite de nombreux échanges effectués. Notons que l'ordre de parcours des rotations défini dans ÉCHANGE est également important : il implique une diminution de temps de vol crédité variant de .3 à 1.8% par rapport à d'autres ordres de parcours.

Au cours de l'expérimentation, nous avons dû apporter quelques modifications à l'algorithme et ajouter deux procédures d'amélioration pour obtenir de meilleurs résultats. Afin d'évaluer notre solution, nous la comparons avec celle d'Air Canada. Nous choisissons pour comparaison leur ensemble de rotations incluant nos vols. Cet ensemble peut contenir des vols supplémentaires étant donné que l'algorithme a été testé sur un sous-ensemble de vols. La comparaison figure dans le tableau suivant, où les temps sont calculés en minutes :

	temps de vol total	temps de vol total crédité	% de temps crédité réellement volé
solution d'Air Canada	17135	23921	71.6%
notre solution	12842	17787.5	72.2%

Nous constatons que nos résultats sont légérement meilleurs que les leurs : nous obtenons un .6% supplémentaire de temps de vol crédité réellement volé.

Rappelons que nous avons dû effectuer l'initialisation des rotations manuellement afin d'éviter que l'algorithme insère dans une rotation un vol nécessaire à une autre. Ainsi, tous les vols ont été examinés pour être regroupés en rotations. Celles-ci ont dû être initialisées non seulement par un vol,

mais par une suite de vols devant faire partie d'une même rotation. En appliquant l'algorithme, nous constatons qu'il n'effectue qu'une Phase3 ramenant les rotations initialisées à leur base. Ainsi, la phase de construction de rotations a été faite en grande partie manuellement, réduisant alors l'algorithme à une application de recherche de plus courts chemins. Nous considérons donc que algorithme d'insertion parallèle basé sur les deux stratégies décrites en section 2.3.1 n'est pas applicable au domaine aérien, dû principalement à la difficulté de trouver une bonne initialisation simple. Rappelons que la principale différence d'application de l'algorithme avec le domaine scolaire se situe précisement à ce niveau : les routes scolaires sont non seulement initialisées par un parcours à effectuer mais également par le chemin aller-retour. Même en essayant de procéder identiquement en initialisant les rotations par un vol aller et un vol retour, le choix de ces vols resterait délicat.

Nous abandonnons donc l'idée d'initialiser les rotations et de les compléter vol après vol, et nous proposons une nouvelle technique de génération des rotations. Nous construirons des suites de vols les plus longues possible afin qu'elles forment des bonnes périodes de service, et par la suite, nous les regrouperons en rotations.

Nous conservons cependant l'idée d'insertion parallèle en ce sens que les suites de vol seront construites parallèlement, étant donné que ce principe s'est avéré efficace pour la construction de rotations équilibrées.

CHAPITRE 3

ALGORITHME PROPOSÉ

Dans ce chapitre, nous proposons un nouvel algorithme de résolution. Nous décrivons ses diverses phases et effectuons son expérimentation sur deux ensembles de données.

3.1 PROCÉDURE

Nous proposons un algorithme qui, pour construire les rotations, se base principalement sur la formation de bonnes périodes de service. Celles-ci sont créées en reliant les vols de manière à minimiser le temps d'attente. L'algorithme se décompose en trois phases principales :

- une phase de construction de segments.
- une phase de formation de rotations.
- une phase d'amélioration.

La première phase crée à partir des vols, des segments ou suites de vol ayant tendance à se lier. Ensuite, la deuxième phase forme à partir de ces segments des périodes de service se regroupant en rotations. Finalement, la dernière phase améliore l'ensemble des rotations obtenues. Nous présentons dans cette section les différentes phases de l'algorithme.

3.1.1 Phase de construction de segments

Cette phase crée à partir de l'ensemble des vols, des segments servant d'élément de base aux rotations. Nous formons les segments avec les vols débutant ou terminant à une ville peu fréquentée(n'ayant pas beaucoup de vols la parcourant). Les vols arrivant à une telle ville sont liés avec leur prochain vol de même avion ou avec le vol qui implique un temps d'attente minimal. En traitant ainsi les vols, nous formons des segments qui se trouvent sûrement dans la solution optimale, étant donné le choix restreint de liaisons de vols à une ville peu fréquentée. La phase de construction de segments, effectuée en deux parties, se décrit en un programme PASCAL de la manière suivante :

```
CRÉATION DE SEGMENTS
première partie
Pour chaque vol terminant à une ville peu fréquentée faire
 début
  chercher le prochain vol de même avion;
  si il existe
   alors créer un segment avec ces deux vols
 fin
```

deuxième partie

Les vols terminant à une ville peu fréquentée,
sans prochain vol de même avion sont non-examinés ;
tantque il existe des vols non-examinés faire
début
déterminer le vol v_1 non-examiné d'heure d'arrivée
minimale à la ville la moins fréquentée ;
créer un segment avec v_1 et le vol impliquant
un temps d'attente minimal ;
v_1 est examiné
fin

Notons que nous ne traitons pas tous les vols selon la deuxième partie afin d'éviter des liens entre un vol v_1 et un vol impliquant un temps d'attente minimal, étant le prochain vol de même avion d'un vol v_j non encore traité. Nous utilisons donc la première partie de la procédure pour effectuer d'abord les liens entre vols de même avion. De plus, dans la dernière partie de la procédure, les villes les moins fréquentées sont traitées en priorité, afin d'y effectuer les meilleurs liens. Remarquons qu'en liant deux à deux tous les vols parcourant les villes peu fréquentées, les segments obtenus peuvent être constitués de plusieurs vols et former déjà des périodes de service ou des rotations.

3.1.2 Phase de formation de rotations

Cette phase crée des rotations en liant les segments entre eux. Elle se déroule en deux étapes principales. La première crée des périodes de service à partir des segments et la deuxième forme des rotations à partir des périodes de service obtenues. Nous décrivons dans cette section chacune de ces étapes.

a) étape de création des périodes de service

L'étape relie ensemble des segments n'impliquant pas de nuit, formant ainsi de périodes de service. Ce processus se réalise en six parties. Les segments multiples (composés de plusieurs vols) sont traités en priorité. Les première et deuxième parties traitent les segments multiples : elles les font suivre du segment débutant avec le prochain vol de même avion, et précéder du segment terminant avec le vol précédant de même avion. La troisième partie relie les segments multiples entre eux. Les quatrième et cinquième parties relient les segments multiples avec des segments quelconques et, finalement, la sixième partie relie les segments quelconques entre eux. Ainsi, nous tentons de compléter les segments déjà composés de plusieurs vols en de bonnes périodes de service.

De plus, les premiers liens effectués sont ceux entre vols de même avion, et les autres sont ceux minimisant le temps d'attente. De cette manière, nous construisons de bons segments en tant que temps d'attente.

Notons également que l'ordre de traitement des segments est important. Nous ajoutons d'abord des segments à la fin des segments d'heure d'arrivée minimale et au début des segments d'heure de départ maximale. Nous évitons ainsi d'obtenir des rotations terminant tôt ou débutant tard leur service, impliquant alors de courtes périodes de service.

Les liens entre segments ne s'effectuent qu'à des villes différentes des bases d'opération. Nous laissons à plus tard la possibilité aux bases d'être non seulement des villes de départ et d'arrivée des rotations, mais également des villes intermédiaires.

Nous décrivons d'abord la forme générale des six parties du processus de la création des périodes de service, puis nous spécifions pour chacune d'elles les critères utilisés :

CRÉATION DES PÉRIODES DE SERVICE

forme générale des parties

Les segments satisfaisant le critère C0 sont non-examinés ;
tantque il existe des segments non-examinés faire
début
 déterminer un premier segment S_1 non-examiné de type T1 ;
 chercher un second segment S_2 de type T2
 si S_2 existe et n'implique pas de nuit
  alors créer un nouveau segment S ;
 S_1 est examiné
fin

première partie

CO : multiples et ne terminant pas à une base.

T1 : -

T2 : débutant avec le prochain vol de même avion du dernier vol de S_1.

S = (S_1, S_2).

deuxième partie

CO : multiples et ne débutant pas à une base.

T1 : -

T2 : terminant avec le vol ayant pour prochain vol de même avion le premier vol de S_1.

S = (S_2, S_1).

troisième partie

CO : multiples et ne terminant pas à une base.

T1 : d'heure d'arrivée minimale.

T2 : multiple et impliquant un temps d'attente minimal.

S : (S_1, S_2).

quatrième partie

CO : multiples et ne terminant pas à une base.

T1 : d'heure d'arrivée minimale.

T2 : quelconque et impliquant un temps d'attente minimal.

S = (S_1, S_2).

cinquième partie

C0 : multiples et ne débutant pas à une base.

T1 : d'heure de départ maximale.

T2 : quelconque et impliquant un temps d'attente minimal.

S = (S_2, S_1).

sixième partie

C0 : quelconques et ne terminant pas à une base.

T1 : d'heure d'arrivée minimale.

T2 : quelconque et impliquant un temps d'attente minimal.

S = (S_1, S_2).

À la suite de ces six parties de l'étape de création de périodes de service, nous obtenons des segments formant des périodes de service ne pouvant se relier entre elles sans impliquer une nuit à une ville différente d'une base. Nous procédons ensuite à l'étape de formation des rotations.

b) étape de formation des rotations

L'étape lie ensemble les segments obtenus (périodes de service) afin de former des rotations. Ce processus se réalise en quatre parties. Nous traitons d'abord, en première et deuxième partie, les segments ne terminant pas à une base. Comme nous l'avons déjà mentionné, ces segments ne peuvent être suivis

par d'autres segments sans impliquer une nuit. Parmi ces segments, ceux qui ne débutent pas non plus à une base sont examinés en priorité en première partie. Ils sont reliés à des segments terminant à une base afin d'éviter l'obtention de rotations composées de nombreuses périodes de service. Nous obtenons ainsi des segments qui débutent et/ou terminent à une base. En deuxième partie, nous traitons les segments qui ne terminent pas à une base(donc qui débutent à une base). Nous les relions également à des segments terminant à une base afin de les ramener le plus tôt à une base. Après ces deux parties, les segments résultant débutent ou terminent à une base. Si chacun d'entre eux ont les mêmes extrémités, alors les rotations sont complétées. Sinon, ceux d'extrémités différentes sont traités en troisième et quatrième partie. En troisième partie, nous essayons de les lier sans nuit avec des segments de bases différentes(étant donné que ceux de même base peuvent former à eux seuls des rotations). En quatrième partie, nous tentons également de les relier sans nuit, cette fois-ci avec des segments de mêmes extrémités. Si cela est impossible, nous les lions alors avec des segments d'extrémités différentes, une nuit étant obligatoire. Notons qu'à la suite des troisième et quatrième parties, les bases peuvent être des villes intermédiaires.

Les liens entre segments se font toujours en minimisant le temps d'attente. Lors d'une nuit, même si le temps crédité(CR) ne dépend pas de la longueur du repos, nous évitons ainsi les situations désagréables où les équipages finissant tôt leur

service, doivent attendre longtemps le lendemain avant de poursuivre leur service.

Nous décrivons les quatre parties de l'étape de formation de rotations sous forme de programme PASCAL :

FORMATION DES ROTATIONS

forme générale des parties

Les segments satisfaisant le critère C0 sont non examinés ;

tantque il existe des segments non-examinés faire

début

déterminer un premier segment S_1 non-examiné

d'heure d'arrivée minimale ;

chercher un second segment S_2 de type T2

si S_2 satisfait le critère C2

alors créer un nouveau segment S

sinon

faire l'action A ;

S_1 est examiné

fin

première partie

C0 : ne touchant pas à une base.

T2 : terminant à une base et débutant le plus tôt.

C2 : existe.

S = (S_1, S_2) avec une nuit obligatoire.

A : -

deuxième partie

C0 : ne débutant qu'à une base.

T2 : terminant à une base et débutant le plus tôt.

C2 : existe.

S = (S_1, S_2) avec une nuit obligatoire.

A : -

troisième partie

C0 : ayant comme extrémités des bases différentes.

T2 : ayant comme extrémités des bases différentes et minimisant le temps d'attente.

C2 : existe et n'implique pas de nuit.

S = (S_1, S_2).

A : -

quatrième partie

C0 : ayant comme extrémités des bases différentes.

T2 : ayant comme extrémités la même base et minimisant le temps d'attente.

C2 : existe et n'implique pas de nuit.

S = (S_1, S_2).

A : trouver un segment S_3 ayant comme extrémités des bases différentes, débutant le plus tôt, et créer un nouveau segment (S_1, S_3) avec une nuit obligatoire.

À la suite des quatre parties de l'étape de formation des rotations, nous obtenons un ensemble de rotations réalisables. Nous procédons alors à la phase d'amélioration.

3.1.3 Phase d'amélioration

Cette phase améliore la solution obtenue à l'aide de quatre procédures d'amélioration. Les deux premières sont celles décrites au chapitre précédant, soit AMÉLIO qui lie des rotations de même base et ÉCHANGE qui échange des vols de période de service lorsqu'il en résulte une amélioration des coûts. La troisième procédure(SEG2) retouche les segments obtenus à la suite de la phase de construction de segments, et la quatrième procédure utilisée(AJOUT) place des vols contenus dans une période de service d'une rotation à la fin ou au début d'une seconde rotation.

Nous décrivons dans cette section ces deux dernières procédures d'amélioration.

a) procédure SEG2

À la suite de la construction de segments, cette procédure échange, lors d'un cas particulier, les liens entre deux segments composés de deux vols. Nous décrivons la procédure de la manière suivante :

SEG2

Les segments composés de deux vols séparés d'une nuit, qui touchent à des bases différentes sont non-examinés ;

tantque il existe des segments non-examinés faire

début

déterminer un premier segment S_1 non_examiné ;

chercher un second segment S_2 composé de deux vols sans nuit, qui touche à une même base, égale à une des deux extrémités de S_1 et qui passe par la même ville intermédiaire que celle de S_1 ;

si S_2 existe et l'échange de lien entre S_1 et S_2 implique que le segment résultant ne touchant pas à la même base, n'ait pas de nuit

alors faire l'échange ;

S_1 est examiné

fin

En échangeant ainsi un vol entre deux segments, la procédure transforme le premier segment de deux vols, ayant comme extrémités des bases différentes et comportant une nuit, en un segment semblable sans nuit. Simultanément, le second segment de deux vols ayant comme extrémités une même base et s'effectuant sans nuit, est transformé en un segment semblable pouvant comporter une nuit. Ainsi, en éliminant le premier segment initial, nous évitons une éventuelle liaison avec un autre segment impliquant une nuit, qui engendrait alors un segment résultant composé d'une période de service intermédiaire d'un seul vol. Une telle liaison est impossible avec la second segment résultant, même s'il comporte une nuit, étant donné qu'il forme déjà une rotation(donc pas de liaisons né-

cessaires). Étant donné que la procédure AMÉLIO ne traite que les premières et dernières périodes de service, il est important d'éviter la formation de périodes de service intermédiaires composées d'un seul vol.

b) procédure AJOUT

Cette procédure traite les rotations qui, après avoir été examinées par les procédures AMÉLIO et ÉCHANGE, ont encore leur première ou dernière période de service P_1 constituée d'un seul vol. Étant donné que ces rotations n'ont pas pu être reliées avec d'autres rotations entières (avec AMÉLIO), la procédure AJOUT tente de les faire précéder ou suivre par une sous-rotation (une suite de vols débutant et terminant à la même base que celle des rotations traitées) contenue dans une période de service P_2. Ainsi, nous prolongeons P_1 et raccourcissons P_2 de la sous-rotation, lorsqu'une amélioration des coûts est entrainée. Celle-ci peut être obtenue directement par le changement effectué ou ultérieurement par un lien possible entre la deuxième rotation et une autre. Dû à cette dernière possibilité, la procédure AMÉLIO doit être appliquée une seconde fois. La procédure AJOUT se décrit de la manière suivante :

AJOUT

Les rotations ayant leur première(ou dernière) période de service composée d'un seul vol sont non-examinés ;

tantque il existe des rotations non-examinées faire

début

déterminer une première rotation R_1 non-examinée ;

chercher une suite de vols S_2 contenue dans une période de service d'une rotation R_2, débutant et terminant à la même base que celle de R_1 ;

si S_2 peut se placer au début(ou à la fin) de R_1

et ce changement implique une diminution des coûts

alors effectuer le changement

sinon si S_2 se trouve au début ou à la fin de R_2

et le reste de R_1 peut se faire suivre ou précéder d'une autre rotation en diminuant les coûts

alors effectuer le changement

R_1 est examinée

fin

La phase d'amélioration tente d'améliorer les segments par la procédure SEG2. Par la suite, elle essaie d'améliorer les rotations obtenues en effectuant d'abord AMÉLIO, puis s'il reste encore des rotations ayant leur première ou dernière période de service composée d'un seul vol, en appliquant AJOUT suivie de AMÉLIO. Rappelons que la procédure ÉCHANGE est utilisée après chaque changement effectué par AMÉLIO.

3.2 EXPÉRIMENTATION

Dans cette section, nous testons l'algorithme proposé sur les données d'Air Canada. Les contraintes utilisées ont déjà été décrites à la section 2.4.

Nous évaluons l'algorithme sur deux ensembles de données : le premier étant composé de vols à fréquence normale et le second de vols avec exception de fréquence.

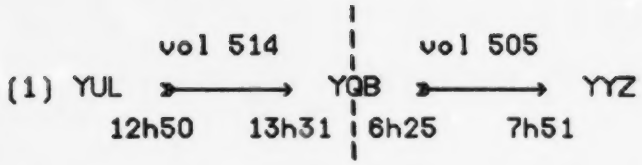
3.2.1 Premier ensemble de données

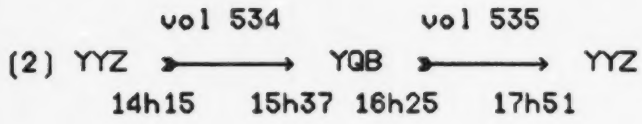
Afin d'évaluer rapidement l'algorithme, nous le testons sur un ensemble de données ne contenant pas de vols avec exception de fréquence. Nous reprenons alors l'ensemble de vols utilisé lors de l'expérimentation de l'algorithme d'insertion parallèle.

Nous obtenons un ensemble de rotations de 17522 minutes de vol crédité, entrainant un pourcentage de 73.3% de temps crédité réellement volé.

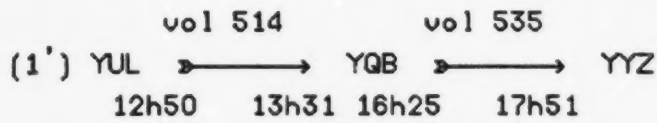
Notons que sans la procédure SEG2, 17696.5 minutes de vol sont créditées, et 72.5% du temps crédité est réellement volé. Nous constatons que SEG2 n'a effectué que le seul échange suivant :

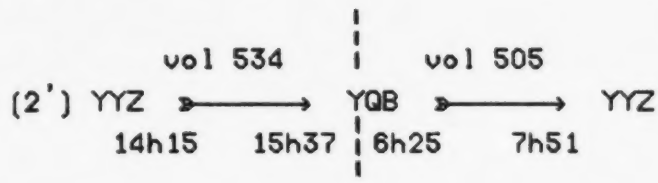
À la ville YQB, la procédure de création de segments a formé entre autre les deux segment suivants :





Le premier segment de bases différentes comporte une nuit, tandis que le second de mêmes bases n'en contient pas. Ces deux segments ont donc été transformé comme suit :





Ainsi, le premier segment résultant de bases différentes ne comporte plus de nuit et le deuxième segment de mêmes bases en contient une. Seuls ces deux liens de vols différents ont entrainé un .8% supplémentaire de temps crédité réellement volé.

La comparaison entre notre solution finale et celle d'Air Canada est donnée dans le tableau suivant, où les temps sont calculés en minutes :

	temps de vol total	temps de vol total crédité	% de temps crédité réellement volé
solution d'Air Canada	17135	23921	71.6%
notre solution	12842	17522	73.3%

Nous obtenons une solution procurant un gain de 1.7% de temps crédité réellement volé. Comme nous l'avons déjà mentionné en section 2.4, l'ensemble de comparaison d'Air Canada contient des vols supplémentaires. Une comparaison plus poussée semble alors superflue.

3.2.2 Deuxième ensemble de données

Afin de pouvoir effectuer une comparaison stricte, nous choisissons pour deuxième ensemble de données, l'ensemble des vols constituant les rotations d'Air Canada affectées à la base YWG. Parmi ces vols, certains ne s'effectuent pas à fréquence normale. Il faut donc ajouter à l'algorithme une procédure traitant les exceptions de fréquence.

Étant donné que les exceptions de fréquence se retrouvent certains jours de chaque semaine, l'unité de base de la construction des rotations n'est plus la journée mais la semaine. Chaque vol effectué x jours de la semaine est donc divisé en x jour-vols ayant leur horaire calculé sur la base d'une semaine. Par exemple, le vol 160 débutant à YYC le mardi à 8h45 est représenté par le jour-vol 160(2) débutant à YYC à 32h45(8h45 +24h).

Nous retrouvons pour vols à exception de fréquence, des vols qui ne s'effectuent que durant la fin de semaine(le samedi et/ou le dimanche) et des vols qui s'effectuent du lundi au vendredi mais non durant toute la fin de semaine(pas le samedi et/ou pas le dimanche). Les exceptions de fréquence se situent donc au niveau de la fin de semaine. Alors, dorénavant, seuls les jour-vols des samedis et des dimanches, correspondant à un vol ne s'effectuant pas tous les jours de la semaine, sont considérés comme jour-vols à exception de fréquence.

À la suite de la phase de construction de segments, nous insérons dans l'algorithme une procédure traitant les segments à exception de fréquence. Ceux-ci sont les segments obtenus ne comportant que des jour-vols à exception de fréquence, ne s'effectuant donc que le samedi et/ou le dimanche.

La procédure, divisée en trois parties, relie ces segments en de nouveaux segments de la manière suivante :

PROCÉDURE D'EXCEPTION DE FRÉQUENCE

première partie

Les segments à exception de fréquence ne terminant pas à une base sont non-examinés ;

tantque il existe des segments non-examinés faire

début

déterminer un premier segment S_1 non-examiné ;

chercher un second segment S_2 à exception de fréquence débutant par le prochain vol de même avion du dernier vol de S_1 ;

si S_2 existe et n'implique pas de nuit

alors créer un nouveau segment $S=(S_1, S_2)$

S_1 est examiné

fin

deuxième partie

Les segments à exception de fréquence ne terminant pas à une base sont non-examinés ;

tantque il existe des segments non-examinés faire

début

déterminer un premier S_1 non-examiné d'heure d'arrivée minimale ;

chercher un second segment S_2 à exception de fréquence impliquant un temps d'attente minimal ;

si S_2 existe et n'implique pas de nuit

alors créer un nouveau segment $S=(S_1, S_2)$

S_1 est examiné

fin

troisième partie

Les segments à exception de fréquence ne terminant pas à une base sont non-examinés ;

tantque il existe des segments non-examinés faire

début

déterminer un premier segment S_1 non-examiné d'heure d'arrivée minimale ;

chercher un second segment S_2 à exception de fréquence impliquant un temps d'attente minimal ;

si S_2 existe

alors créer un nouveau segment composé de S_1 et S_2 séparés par une nuit obligatoire ;

S_1 est examiné

fin

Ainsi, les segments à exception de fréquence se lient entre eux en minimisant le temps d'attente(ou de repos) impliqué. La procédure d'exception de fréquence est non seulement utile au regroupement des vols à exception de fréquence, mais , de ce fait, évite des liaisons déplaisantes lors de la formation des rotations(par exemple, un lien entre un vol ne s'effectuant que le samedi et un second débutant un lundi).

En appliquant l'algorithme muni de la procédure d'exception de fréquence sur notre deuxième ensemble de vols, nous obtenons un ensemble de rotations de 34381 minutes de vol crédité pour 26904 minutes de vol effectué.

Notons que de nombreux vols parmi les 315 jour-vols de notre ensemble débutent ou terminent à la base YWG. De ce fait, celle-ci peut facilement servir de ville intermédiaire. Or, étant donné que certains vols terminant à YWG ont pour prochain vol de même avion un vol quittant YWG dans la même journée, nous allons tester l'algorithme en permettant à la base d'être une ville intermédiaire lors de la première et deuxième partie de l'étape de création de périodes de service. Les critéres "les segments multiples ne terminant pas à une base sont non-examinés" et "les segments multiples ne débutant pas à une base sont non-examinés" correspondant respectivement à la première et deuxième partie, deviennent donc simplement : "les segments multiples sont non-examinés". Ainsi, nous permettons la liaison des vols terminant à YWG avec leur vol de même avion lorsqu'elle n'implique pas de nuit, durant la création de périodes de service. Avec ce léger changement, nous obtenons un ensemble de 34223 minutes de vol crédité, soit une diminution de 158 minutes de vol crédité.

Nous comparons notre ensemble final de rotations avec celui d'Air Canada obtenu à partir du même ensemble de vols desservant YWG. Les pourcentages de temps crédité sont donnés dans le tableau suivant, où les temps sont calculés en minutes :

	temps de vol total	temps de vol total crédité	% de temps crédité réellement volé
solution d'Air Canada	26904	34732	77.5%
notre solution	26904	34223	78.6%

Nous obtenons donc une solution apportant un gain de 1.1% de temps crédité volé par rapport à la solution d'Air Canada.

Il peut ête également intéressant d'effectuer une comparaison des rotations en tant que type de périodes de service obtenues(nombre de nuits nécessaires, nombre de périodes par rotation, etc). Pour cela, examinons le tableau à la page suivante.

Nous constatons qu'avec deux rotations et trois périodes de service en moins, nous n'avons sauvé qu'une seule nuit mais nous avons formé de meilleures périodes de service. En effet, sur nos 90 périodes de service, 41 sont de bonnes périodes(où le temps crédité est égal au temps volé) tandis que sur leurs 93 périodes de service, seulement 38 sont bonnes. Nous avons alors sauvé 509 minutes de temps crédité non volé en formant de bonnes périodes de service, ce qui représente un coût non négligeable.

	solution d'Air Canada	notre solution
nombre de rotations débutant un jour de la semaine	43	41
nombre de périodes de service	93	90
nombre de périodes de repos (nombre de fois où le critère CR est appliqué)	50	49
nombre de périodes de service où le critére MVL ou TP/λ_{TP} est appliqué	55	49
nombre de minutes créditées non volées	7828	7319

Le programme de l'algorithme complet est donné en annexe. Notons qu'il peut certainement être rendu plus efficace, mais nous avons préféré concentrer notre attention sur la recherche

d'une bonne solution plutôt que sur l'optimisation du code. Nous pensons que celle-ci devrait intervenir lors de son implantation éventuelle.

Le programme utilise actuellement 21 tableaux de dimension égale au nombre de vols(nbvol), 1 tableau de dimension égale au nombre d'éléments d'une rotation(nbelrot), 1 matrice de rotations de dimension (nbvol x nbelrot) et 3 tableaux de dimension (nbvol x nombre d'éléments d'une période de service); soit un total de (96 nbvol + 30) avec nbelrot=30 et nombre d'éléments d'une période de service=15.

Le temps de calcul du programme pour notre ensemble de vols à exception de fréquence comportant 315 vols est d'environ 115 secondes.

CHAPITRE 4

CONCLUSION

Nous avons atteint le but que nous nous étions fixé en développant un algorithme heuristique simple qui nous a fourni un bon ensemble de rotations.

Certes, avant de penser à une implantation, des tests supplémentaires devraient être réalisés afin de vérifier la stabilité de l'algorithme et d'effectuer quelques ajustements. Nous pourrions, par exemple, expérimenter l'algorithme sur chaque ensemble de vols DC9 constituant les rotations d'Air Canada affectées respectivement aux bases YYZ et YUL (celui correspondant à la base YWG étant déjà testé) ou sur l'ensemble global contenant tous les vols DC9. Nous pourrions aussi tester l'algorithme sur un ensemble de vols d'un type d'avion différent (DC8, L1011,...).

Une analyse de sensitivité pourrait être également envisagée afin d'évaluer par exemple l'impact d'une modification de contrainte, d'une élimination d'un vol, ou d'un ajoût d'une base sur le coût de la solution.

De plus, il serait intéressant de poursuivre cette étude en comparant notre méthode avec d'autres. Nous pourrions par

exemple utiliser notre solution comme initialisation pour la méthode de Rubin utilisée actuellement à Air Canada, afin d'évaluer une éventuelle amélioration. Nous pourrions également l'utiliser comme solution de départ pour l'algorithme de Crainic ou la comparer directement avec la sienne.

Une expérience supplémentaire tout aussi intéressante serait d'appliquer notre algorithme sur un réseau relativement petit pour pouvoir évaluer la proximité de notre solution à la solution optimale en résolvant exactement le problème de recouvrement d'ensemble.

RÉFÉRENCES

(1) E.Baker, L.Bodin, W.Finnegan, R.Ponder, "Efficient Heuristic Solutions to an Airline Crew Scheduling Problem.", AIIE Transactions, vol.11, No2, p.79-85, 1979.

(2) E.Balas, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables.", Operations Research 19, p.517-546, 1965.

(3) E.Balas "Intersection Cuts - An New Type of Cutting Planes, for Integer Programming.", Operations Research 19, p.19-39, 1971

(4) E.Balas, A.Ho, "Set Covering Algorithms Using Cutting Planes, Heuristics and Subgradient Optimization : A Computational Study.", Mathematical Programming, vol.12, p.37-60, 1980.

(5) M.Bellmore, H.-D.Ratcliff, "Set Covering and Involutory Bases.", Management Science, vol.18, p.194-206, 1971.

(6) L.Chapleau, J.Ferland, G.Lapalme, J.-M.Rousseau, "A Parallel Insert Method for the Capacitated Arc Routing Problem.", Publication #234, Centre de recherche sur les transports, 1982.

(7) T.Crainic, "Le problème des horaires d'équipage d'une compagnie d'aviation.", Publication #122, Centre de recherche sur les transports, 1978.

(8) J.Etcheberry, "The Set Covering Problem : A New Implicit Enumeration Algorithm.", Operations Research, vol.25, p.760-772, 1977.

(9) R.-S.Garfinkel, G.-L.Nemhauser, "The Set Partitioning Problem : Set Covering with Equality Constraints.", Operations Research, vol.17, p.848-856, 1969.

(10) A.-M. Geoffrion, "Integer Programming by Implicit Enumeration and Balas Method.", Siam Review, vol.9, p.178-190, 1967.

(11) F.Glover, "A New Foundation for a Simplified Integer Programming Algorithm.", Operations Research, vol.16, p.727-740, 1968.

(12) F.Glover, "Surrogate Constraints.", Operations Research, vol.16, p.741-749, 1968.

(13) R.-E.Gomory, "An All-Integer Programming Algorithm.", in Industrial Scheduling, Muth & Thomson(eds), Prentice-Hall, 1963.

(14) E.Heurgon, "Un problème de recouvrement : l'habillage des horaires d'une ligne d'autobus.", Revue francaise d'automatique, d'informatique et de recherche opérationnelle, vol.6, p.13-29, 1972.

(15) C.-E. Lemke, H.-M. Salkin, K.Spielberg, "Set Covering by Single Branch Enumeration with Linear Programming Subproblems.", Operations Research, vol.19, p.998-1022, 1971.

(16) R.Marsten, "An Algorithm for Large Set Partitioning Problems.", Management Scince, vol.20, p.774-787, 1974.

(17) J.-F.Pierce, "Application of Combinatorial Programming to a Class of All-Zero-One Integer Programming Problems.", Management Science, vol.15, p.191-209, 1968.

(18) J.-F.Pierce, J.-S.Lasky, "Improved Combinatorial Programming Algorithms for a Class of All-Zero-One Integer Programming Problems.", Management Science, vol.19, p.528-543, 1973.

(19) J.Rubin, "A Technique for the Solution of Massive Set Covering Problems, with Applications to an Airline Crew Scheduling.", Transportation Science, vol.7, p.34-48, 1973.

(20) H.-M.Salkin, R.-D.Koncal, "Set Covering by an All Integer Algorithm : Computational Results.", Journal of the Association of Computing Machinery, vol.20, p.189-193, 1973.

(21) J.-F.Shapiro, "Group Theoretic Algorithms for the Integer Programming Problem. II : Extension to a General Algorithm", Operations Research, vol.16, p.928-947, 1968.

(22) H.Thiriez, "Airline Crew Scheduling : A Group Theoretic Approach", Ph.D.Thesis, Department of Aeronautics, M.I.T., 1969.

(23) R.-D.Young, "A Simplified Primal(All-Integer) Integer Programming Algorithm", Operations Research, vol.16, p.750-782, 1968.

REMERCIEMENTS

Je désire remercier Monsieur Jean-Marc Rousseau, mon directeur de thèse, pour ses suggestions et son support tout au long de ce stage.

Je désire également remercier Lise Fournelle ainsi que Luc Chapleau pour leur collaboration.